



Incorporation of Time Delayed Measurements in a Discrete-time Kalman Filter

Larsen, Thomas Dall; Andersen, Nils Axel; Ravn, Ole; Poulsen, Niels Kjølstad

Published in:
Decision and Control, 1998. Proceedings of the 37th IEEE Conference on

Link to article, DOI:
[10.1109/CDC.1998.761918](https://doi.org/10.1109/CDC.1998.761918)

Publication date:
1998

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Larsen, T. D., Andersen, N. A., Ravn, O., & Poulsen, N. K. (1998). Incorporation of Time Delayed Measurements in a Discrete-time Kalman Filter. In *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on* (Vol. 4, pp. 3972-3977). IEEE. <https://doi.org/10.1109/CDC.1998.761918>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Incorporation of Time Delayed Measurements in a Discrete-time Kalman Filter

Thomas Dall Larsen, Nils A. Andersen & Ole Ravn
Department of Automation, Technical University of Denmark
Building 326, DK-2800 Lyngby, Denmark
e-mail: {tdl, naa, or}@iau.dtu.dk

Niels Kjølstad Poulsen
Department of Mathematical Modelling, Technical University of Denmark
Building 321, DK-2800 Lyngby, Denmark
e-mail: nkp@imm.dtu.dk

Abstract

In many practical systems there is a delay in some of the sensor devices, for instance vision measurements that may have a long processing time. How to fuse these measurements in a Kalman filter is not a trivial problem if the computational delay is critical. Depending on how much time there is at hand, the designer has to make trade offs between optimality and computational burden of the filter. In this paper various methods in the literature along with a new method proposed by the authors will be presented and compared. The new method is based on "extrapolating" the measurement to present time using past and present estimates of the Kalman filter and calculating an optimal gain for this extrapolated measurement.

1 Introduction

This paper considers the problem of designing discrete-time Kalman filters to systems where some results of the measurements are delayed. Most of the work that has been done prior in this field considers only filters where no measurements has been fused in the delay period (i.e. the time period from the measurement is taken till it is available), see for instance [1]. In many applications, especially in the field of autonomous vehicles, however, the Kalman filter will fuse measurements from faster sensors in this delay period. Typically the delayed measurements will origin from a vision system and the fast measurements from a sonar or dead reckoning system.

Other authors have focused on systems, where system and output equations have a common delay, as in [2] and in [3]. Kalman filters where the output is delayed only a fraction of the sample time can be handled in

an optimal manner by modifying the output equation of the filter as shown in [4].

Delays consisting of a small number of samples can be handled optimally in the discrete-time Kalman filter by augmenting the state vector accordingly - see for instance [5] or [6]. As the system order increases with the delay size, however, this method is mostly used when the time delay is a small number of samples, as the computational burden otherwise may become undesirably high.

If only a few measurements are fused in the delay period or if the computational burden of the filter is uncritical, an optimal filter estimate incorporating the delayed measurement can be obtained simply by recalculating the filter through the delay period. As this often becomes too time consuming in practical systems, a method is derived in [7] where it suffices to calculate a correction term and add this to the filter estimate when the delayed measurement arrives. This method is optimal in certain time intervals under certain conditions and will be described in this paper along with a modification introduced by the authors that extends these periods of optimality.

Furthermore, a new method that does not guarantee optimality under all conditions but is useful in many practical systems will be described. The new method, based on *extrapolating* the delayed measurements, is especially suited for the type of filtering problems mentioned above, where a number of (imprecise) measurements has been fused in the delay period. Regardless of the number and nature of the measurements this method is a simple and computationally cheap way of accounting for delays.

2 System and Filter Equations

A linear discrete system observed by non-delayed measurements where both process and measurements are influenced by additive Gaussian noise can be put in state space form as follows:

$$x_{k+1} = A_k x_k + B_k u_k + w_k, \quad (1)$$

$$z_k = C_k x_k + v_k, \quad (2)$$

where: $w_k \sim N(0, Q_k)$ and $v_k \sim N(0, R_k)$.

Without loss of generality we can assume that the two noise sources v_k and w_k are independent ($E[w_i v_j^T] = 0$). The optimal state estimator minimizing the variances of the estimation error will then be a Kalman filter. A proof for this, along with a derivation of the Kalman filter equations can be found in [8]. The equations are summarized below in (3) to (7):

$$\hat{x}_{k+1} = A_k \hat{x}_k(+) + B_k u_k \quad (3)$$

$$P_{k+1} = A_k P_k(+) A_k^T + Q_k \quad (4)$$

$$K_k = P_k C_k^T [C_k P_k C_k^T + R_k]^{-1} \quad (5)$$

$$\hat{x}_k(+) = \hat{x}_k + K_k [z_k - C_k \hat{x}_k] \quad (6)$$

$$P_k(+) = [I - K_k C_k] P_k \quad (7)$$

If the system (1) furthermore has an output that is delayed N samples, for instance due to a slow sensor or a long processing time of the sensor data, there will be a second output equation:

$$z_k^* = C_s^* x_s + v_k^*, \quad v_k^* \sim N(0, R_k^*), \quad (8)$$

where $s = k - N$.

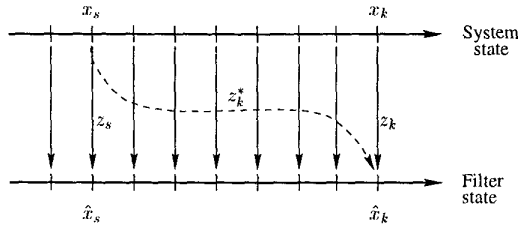


Figure 1: System with an N sample delayed output.

The delayed measurement cannot be fused using the normal Kalman filter equations but requires some modifications in the structure of the filter.

3 Incorporating Delayed Measurements

As mentioned in section 1 a number of different methods has been proposed for incorporating delayed measurements in the Kalman filter. The system defined in

section 2, however, has to the authors' knowledge only been treated in [7]. The method proposed by Alexander will in some cases be highly suited to the type of systems considered in this paper and will therefore be summarized in this section along with a modification suggested by the authors. Subsequently, a new method which can be used when Alexander's method comes to short, will be presented.

3.1 Updating Covariance and State at Different Times

Using the standard Kalman filter equations, the measurement z_k^* should be fused at time s , causing a correction in the state estimate and a decrease in the state covariance. As the state covariance matrix decides the Kalman gain, the measurements occurring after this will all be fused differently than if the measurement update for z_k^* is omitted. If therefore the measurement z_k^* is delayed N samples and fused at time k , the data update should reflect the fact that the N data updates from time s to k , and therefore the state and covariance estimates, have all been affected by the delay in a complex manner.

Equations that account for this when fusing z_k^* at time k has been derived in [7] but are of such complexity that they in many cases are not feasible¹. It is therefore suggested that if the measurement sensitivity matrix, C_s^* , and the noise distribution matrix, R_k^* , is known at time s , the filter covariance matrix should be updated as if the measurement is available. This leads the measurements in the delay period to be fused as if z_k^* had been fused at time s . At time k , when z_k^* is available, incorporating z_k^* is then greatly simplified, by adding the following quantity after z_k has been fused:

$$\delta \hat{x}_k = M_* K_s (z_k^* - C_s^* \hat{x}_s) \quad (9)$$

If the delay is zero, M_* is the identity matrix. For $N > 0$, M_* is given by:

$$M_* = \prod_{i=0}^{N-1} (I - K'_{k-i} C_{k-i}) A_{k-i-1} \quad (10)$$

The prime on K' signifies that these Kalman gain matrices have been calculated using a covariance matrix updated at time s with the covariance of the delayed measurement. As one factor in the product above can be calculated at each sample time the method only requires two matrix multiplications at each sample time.

The method implies that the covariance of the filter will be wrong in a period of N samples leading measurements in this period to be fused suboptimally. However, after the correction term in (9) is added, the filter state and covariance will once again be optimal.

¹In fact the computational complexity of these equations is comparable to recalculating the Kalman filter through the delay of N samples.

3.1.1 A Modification to Ensure Optimality:

As mentioned in section 3.1 the filter estimates will be suboptimal from the time the measurement z_k^* is taken till it is fused. This can cause problems, especially if data validation is performed using the incorrect P -matrix leading the filter to discard valid measurements.

At the expense of some additional computations this problem can be avoided by making the filter estimate optimal at all times. This is done by starting up an extra filter at time s that uses the covariance R_k^* as suggested by [7] and running this in parallel to the optimal filter. Up until time k , the estimates from the optimal filter will be used but at time k when the delayed measurement arrives, this will be fused in the parallel filter and this filter will now have the optimal state estimate.

This simple modification of the method guarantees optimality at all times but imposes twice the computational burden in the delay time period (from s to k).

3.2 Extrapolating the Measurement

The method described in section 3.1 required the measurement sensitivity matrix, C_s^* , and the noise distribution matrix, R_k^* , to be known at time s . In many cases they are not. If for instance the measurement is a vision measurement, the uncertainty of the measurement will often be unknown until the data is processed as it depends on the relative positioning of the camera and object. Similarly C_s^* may depend on positioning and occlusions and therefore also not be known until the data is processed. A method that does not require knowledge about z_k^* until time k is therefore needed.

For non-delayed measurements, the residual used to calculate the new estimate, is defined by:

$$\epsilon_k = z_k - C_k \hat{x}_k \quad (11)$$

When the N -samples delayed measurement given in (8) arrives at time k , the filter and measurement state will relate to different times and a residual relating to time k cannot be attained. But if the filter state from time s has been stored, the residual that would have been used at time s if the measurement had not been delayed can be used in the update at time k :

$$\epsilon_k := \epsilon_s = z_k^* - C_s^* \hat{x}_s \quad (12)$$

This is equivalent of extrapolating the measurement, z_k^* , to a present measurement, z_k^{int} :

$$z_k^{int} = z_k^* + C_k^* \hat{x}_k - C_s^* \hat{x}_s, \quad (13)$$

and fusing this at time k using the ordinary residual defined in (11).

The extrapolated measurement is given by:

$$z_k^{int} = z_k^* + C_k^* \hat{x}_k - C_s^* \hat{x}_s$$

$$\begin{aligned} &= C_s^* x_s + v_k^* + C_k^* \hat{x}_k - C_s^* \hat{x}_s \\ &= C_k^* x_k + v_k^* + C_k^* \tilde{x}_k - C_s^* \tilde{x}_s \end{aligned} \quad (14)$$

$$= C_k^* x_k + v_k^{int} \quad (15)$$

where the estimation error $\tilde{x} = \hat{x} - x$.

This new extrapolated measurement is seen to have the standard form as in (2), except that here there is a correlation between the noise process v_k^{int} and the state x_k . The optimal gain for fusing z_k^{int} and the resulting filter covariance decrease will now be derived.

If the measurement is fused using the data update in (6) with an arbitrary gain K_k , the estimation error, $\tilde{x}_k(+)$, becomes:

$$\tilde{x}_k(+) = (I - K_k C_k^*) \tilde{x}_k + K_k v_k^{int} \quad (16)$$

The variance of the estimation error is:

$$\begin{aligned} P_k(+) &= E\{\tilde{x}_k(+) \tilde{x}_k^T(+)\} \\ &= (I - K_k C_k^*) P_k (I - K_k C_k^*)^T \\ &\quad + (I - K_k C_k^*) E\{\tilde{x}_k v_k^{intT}\} K_k^T \\ &\quad + K_k E\{v_k^{int} \tilde{x}_k^T\} (I - K_k C_k^*)^T \\ &\quad + K_k E\{v_k^{int} v_k^{intT}\} K_k^T \end{aligned} \quad (17)$$

Let: $M = E\{\tilde{x}_s \tilde{x}_k^T\}$. The covariances in (17) can then be found from (14):

$$E\{\tilde{x}_k v_k^{intT}\} = P_k C_k^{*T} - M^T C_s^{*T} \quad (18)$$

$$\begin{aligned} E\{v_k^{int} v_k^{intT}\} &= R_k^* + C_k^* P_k C_k^{*T} + C_s^* P_s C_s^{*T} \\ &\quad - C_s^* M C_k^{*T} - C_k^* M^T C_s^{*T} \end{aligned} \quad (19)$$

Inserting (18) and (19) in (17) and rearranging the terms leads to:

$$\begin{aligned} P_k(+) &= P_k - M^T C_s^{*T} K_k^T - K_k C_s^* M \\ &\quad + K_k C_s^* P_s C_s^{*T} K_k^T + K_k R_k^* K_k^T \end{aligned} \quad (20)$$

The gain, K_k , should be chosen so that the variances of the estimation errors are as low as possible. This is obtained when:

$$\frac{\partial \text{trace}(P_k(+))}{\partial K_k} = 0$$

Differentiating (22) and isolating K_k yields:

$$K_k = M^T C_s^{*T} [C_s^* P_s C_s^{*T} + R_k^*]^{-1} \quad (21)$$

This is therefore the optimal gain for fusing the measurement z_k^{int} . Inserting (21) in (20) leads to a simpler form for $P_k(+)$:

$$P_k(+) = P_k - K_k C_s^* M \quad (22)$$

The covariance matrix, M , can be found by observing the propagation of the estimation error \tilde{x} . Through the time update (3), \tilde{x} becomes:

$$\tilde{x}_{k+1} = A_k \tilde{x}_k(+) - w_k$$

After the data update (6), \tilde{x} is:

$$\tilde{x}_k(+) = (I - K_k C_k) \tilde{x}_k + K_k v_k$$

Through N succeeding time and data updates from time s to k , the estimation error therefore becomes:

$$\begin{aligned} \tilde{x}_k(+) &= \prod_{i=0}^{N-1} [(I - K_{k-i} C_{k-i}) A_{k-i-1}] \tilde{x}_s(+) \\ &\quad + f_1(w_s \cdots w_{k-1}) + f_2(v_{s+1} \cdots v_k) \end{aligned}$$

Where f_1 and f_2 are functions of the noise sequences v and w . As \tilde{x}_s is uncorrelated with these noise sequences, the covariance, M , becomes:

$$\begin{aligned} M &= E\{\tilde{x}_s \tilde{x}_k^T\} \\ &= P_s \prod_{i=0}^{N-1} A_{s+i}^T (I - K_{s+i+1} C_{s+i+1})^T \\ &= P_s M_s^T, \end{aligned} \quad (23)$$

where:

$$M_s = \prod_{i=0}^{N-1} (I - K_{k-i} C_{k-i}) A_{k-i-1} \quad (24)$$

Observe that the correction term, M_s , is similar to (10) in section 3.1 except that in (10) the Kalman gains reflect the z_k^* data update at time s .

Substituting (23) in (21) and (22) yields:

$$P_k(+) = P_k - K_k C_s^* P_s M_s^T \quad (25)$$

and:

$$K_k = M_s P_s C_s^{*T} [C_s^* P_s C_s^{*T} + R_k^*]^{-1} \quad (26)$$

Notice that when the delay N is zero, $M_s = I$ and the Kalman gain equation (26) reduces to the standard form (5) and the covariance update (25) reduces to (7).

So by calculating an extrapolated measurement, z_k^{int} , using (13) the delayed measurement can be fused in a simple and computationally cheap manner using equation (24) - (26).

3.2.1 Optimality of Extrapolation: Observe that though the gain suggested in (26) is statistically optimal for fusing the extrapolated measurement, the extrapolation method *itself* is still not optimal. In order for the method to be optimal the Kalman gains in (24) should reflect a data update at time s , that is equation (10) should equal (24).

If no measurements has been fused in the delay period equation (24) and (10) are identical and the extrapolation will be optimal. This important case is quite common in praxis, for instance when a dead reckoning

system is used as the system model. This is a popular Kalman filter type on mobile robots, see for instance [9] or [10]. Here:

$$M_s = \prod_{i=0}^{N-1} A_{k-i-1}$$

The matrix A is a linearized system matrix as described in [10]. If the movement of the robot is assumed linear the output from the dead reckoning system can be transformed to a linear and angular displacement Δd_k and $\Delta \theta_k$. The robot coordinates in a global coordinate frame can then be updated by (see [11]):

$$\begin{aligned} X_{k+1} &= X_k + \Delta d_k \cos(\theta_k + \frac{\Delta \theta_k}{2}) \\ Y_{k+1} &= Y_k + \Delta d_k \sin(\theta_k + \frac{\Delta \theta_k}{2}) \\ \theta_{k+1} &= \theta_k + \Delta \theta_k \end{aligned}$$

The A matrix is seen to be:

$$\begin{aligned} A &= \begin{bmatrix} 1 & 0 & a_{13} \\ 0 & 1 & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \\ a_{13} &= \frac{\partial X}{\partial \Delta \theta} = -\Delta d \sin(\theta + \frac{\Delta \theta}{2}) \\ a_{23} &= \frac{\partial Y}{\partial \Delta \theta} = \Delta d \cos(\theta + \frac{\Delta \theta}{2}) \end{aligned}$$

Considering the structure of A , the matrix M^* can be found by:

$$M_s = \begin{bmatrix} 1 & 0 & \sum_{i=0}^{N-1} a_{13,i} \\ 0 & 1 & \sum_{i=0}^{N-1} a_{23,i} \\ 0 & 0 & 1 \end{bmatrix}$$

When this particular filter type is used, the extrapolation method therefore provides a very simple and optimal² way of accounting for delays.

4 Example

Consider now, as a continuous plant a typical DC motor with the shaft position, $\theta(t)$, as output and the anchor voltage, $u(t)$ as input. A potentiometer and a camera both observe the shaft position. The potentiometer measurement is continuous and noisy and the vision measurement is discrete, delayed but accurate. Both the process and the measurements are influenced by independent Gaussian white noise processes. A state space formulation of the plant is given below:

$$\begin{aligned} \dot{x}(t) &= A_c x(t) + B_c u(t) + w(t) \\ &= \begin{bmatrix} 0 & 1 \\ 0 & -\omega_m \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ k_m \end{bmatrix} u(t) + w(t) \\ z(t) &= [1 \ 0] x(t) + v(t) \\ z^*(kT) &= [1 \ 0] x(kT - t_d) + v^*(kT), \quad k \in \mathcal{N} \end{aligned}$$

²Strictly speaking the filter is not optimal as the system is nonlinear.

The distribution of the process noise is:

$$w(t) \sim N(0, \begin{bmatrix} 0 \\ Q \end{bmatrix}),$$

and the distribution of the measurement noises are:

$$v(t) \sim N(0, R), v^*(kT) \sim N(0, R^*)$$

The plant is discretized and a discrete Kalman filter is designed as described in section 2.

This plant is now simulated using different methods to account for the delayed measurement, namely by:

A Recalculating the filter when z_k^* arrives

B Using Alexander's method

C Using the modified version of Alexander's method

D Extrapolating the measurement

The variances of the potentiometer noise is $R = 10^{-3}$ while the proces noise, Q , and the vision noise, R^* , are varied in the simulation.

Table 1 shows the averaged variances of the estimation error, $e = y - \hat{y}$, from Monte Carlo simulations using the four methods above and using different values for the noise variances and the initial estimation error. The variances are normalized with the results from a simulation with no delay on the vision measurements. The

Q	R^*	A	B	C	D
0	10^{-5}	1.16	1.48	1.16	1.35
0	10^{-4}	1.20	1.23	1.20	1.37
10^{-4}	10^{-5}	1.34	1.66	1.34	1.51
10^{-4}	10^{-4}	1.10	1.11	1.10	1.18

Table 1: Normalized estimation error variance

normalized variances in table 1 are all higher than one, meaning that no method fully compensates for the delay (so rather unsurprisingly we would prefer that the measurement was not delayed). Also it is observed that the modified Alexander as expected yields the same results as recalculation as both these methods are optimal.

It is also seen that although Alexander's method is optimal in the time intervals outside the delay period and the extrapolation method here is suboptimal always, it is not obvious which of the two methods performs best. In the two simulations with low vision measurement noise the extrapolation yields the lowest variance and in the other two simulations Alexander's method performs best. It is clear that comparisons between

these two methods should be done with some caution as the relative performance of the methods changes in different conditions.

In comparing the filters the computational burden imposed by the filters should also be considered. Table 2 shows the amount of floating point operations in the different filters, scaled with respect to a filter that fuses an undelayed vision measurement at time k . Though these numbers are illustrative, of course the absolute as well as the relative size strongly depends on the specific system, especially the system order.

Sample	A	B	C	D
s	1.0	1.4	1.4	1.0
$s \rightarrow k$	1.0	1.3	2.3	1.1
k	N	1.0	1.7	1.1

Table 2: Normalized computational burden

It is obvious that recalculation can only be used if the delay N is small or if the computation time is uncritical. If the measurement variance and sensitivity matrix are known when the measurement is taken the modified Alexander yields exactly the same results with less computation. Both the extrapolation method and Alexander's method are even cheaper computationally but does not guarantee optimality at all times.

5 Conclusion

In this paper a new method for incorporating measurements with delays of arbitrary size in a Kalman filter has been introduced. The method is fast and can be applied to a wide variety of systems, but does not guarantee optimality under all conditions. If the covariance and the measurement sensitivity matrix of the delayed measurement is at hand at the time where the measurement is taken, a different method introduced in [7] and modified in this paper will give an optimal and fairly fast estimate.

It was shown that if no measurements are fused in the delay period, the extrapolation method will be optimal. If an odometric filter type is used where for instance encoder readings are used as a system model, the extrapolation method will also be optimal and the algorithm very simple.

References

- [1] Stelios C. A. Thomopoulos. Decentralized filtering in the presence of delays: Discrete-time and continuous-time case. *Information Sciences*, 1994.

- [2] G. N. Mil'shtein and S. A. P'yanzin. A discretization method in designing an optimal filter for systems with delay. In *Automation and Remote Control*, 1990.
- [3] Sang Jeong Lee, Seok min Hong, and Graham C. Goodwin. Loop transfer recovery for linear systems with delays in the state and the output. *International Journal of Control*, 1994.
- [4] Gert L. Andersen, Anders C. Christensen, and Ole Ravn. Augmented models for improving vision control of a mobile robot. In *3rd IEEE Conference on Control Applications*, 1994.
- [5] Feng-Hsiag Hsiao and Shing-Tai Pan. Robust kalman filter synthesis for uncertain multiple time-delay stochastic systems. *Journal of Dynamic Systems, measurement, and control*, 1996.
- [6] E. Kaszkurewicz and A. Bhaya. Discrete-time state estimation with two counters and measurement delay. In *Proceedings of the 35th IEEE Conference on Decision and Control*, 1996.
- [7] Harold L. Alexander. State estimation for distributed systems with sensing delay. In *SPIE vol. 1470 Data Structures and Target Classification*, 1991.
- [8] Arthur Gelb. *Applied Optimal Estimation*. The analytical Sciences Corporation, 1974.
- [9] F. Chenavier and J. L. Crowley. Position estimation for a mobile robot using vision and odometry. In *Proceedings of the 1992 IEEE International Conference on Robotics And Automation*, Nice, France, 1992.
- [10] Satoshi Murata and Takeshi Hirose. Onboard locating system using real-time image processing for a self-navigating vehicle. *IEEE Transactions on Industrial Electronics*, 40(1), February 1993.
- [11] C. M. Wang. Location estimation and uncertainty analysis for mobile robots. In *Proceedings of the 1988 International Conference on Robotics and Automation*, 1988.